ブラウザのUIのバグを探す

2017/11/25-26 セキュそば勉強会#40 戸隠

Masato Kinugawa

自己紹介

名前：Masato Kinugawa
所属：Cure53
趣味：音楽鑑賞とXSS

話すこと

最近いくつか報告したブラウザのUIのバグを
どのように発見したかをお話します。

今回発見したのはただのバグですが、
発想は脆弱性の発見にも応用できるはず。

技術的な話はあまりしません！

UIのバグ/脆弱性

= 見た目や操作に関連するもの、例えば：

- アドレスバー偽装
- 特権的操作の確認ダイアログに対するクリックジャッキング

(脆弱性未満の問題)

- ポップアップブロッカーのバイパス など

今回扱うバグ(脆弱性でない)

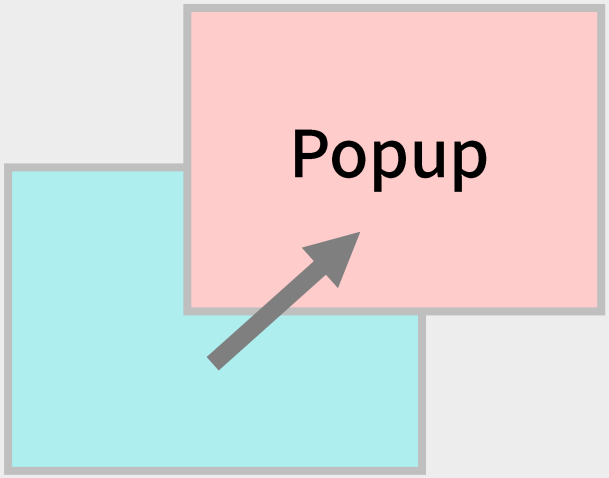
- Popunder Preventerのバイパス

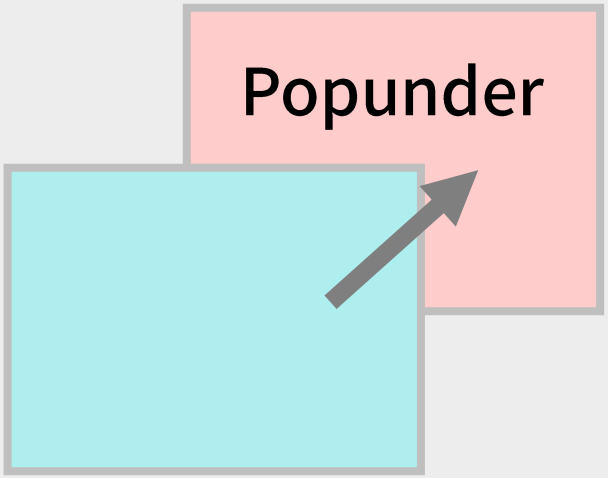
Popunder Preventer？

- Chromeに実装されているポップアンダーウインドウの作成を防止する機能
- ソースのファイル名から名前を拝借

https://cs.chromium.org/chromium/src/chrome/browser/ui/blocked_content/**popunder_preventer.cc**

ポップアップとポップアンダー





➡親ウインドウの裏に新しいウインドウが
開かれるのがポップアンダー

Preventerの役割

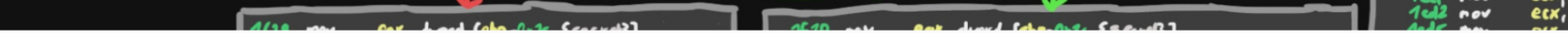
- 新しく開かれたウインドウを常に前へ
- 迷惑な広告を阻止する目的(だと思う)

Chromeではもはや単純にフォーカスをJSから操作して
Popunderを作ることはできない：

```
newWin = window.open('//example.com/','w','a');
newWin.blur();//新しいウインドウのフォーカスを離す
window.focus();//親にフォーカスを移す
```

興味を持ったきっかけ

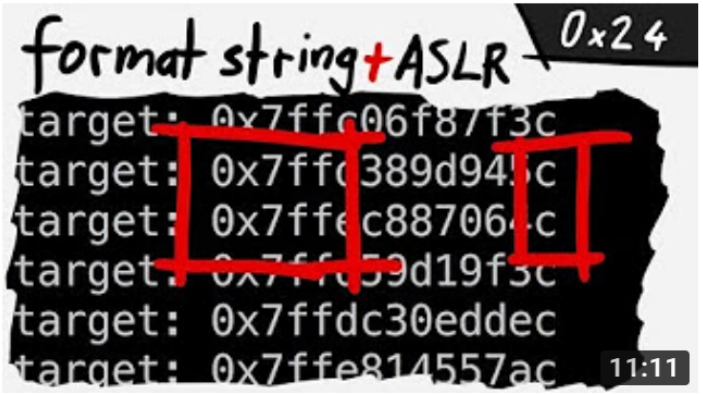
- @LiveOverflow さんの動画で知る
  - バイナリやWeb、CTFの解説など、セキュリティ関係の動画をアップしているYouTubeチャンネル



LiveOverflow

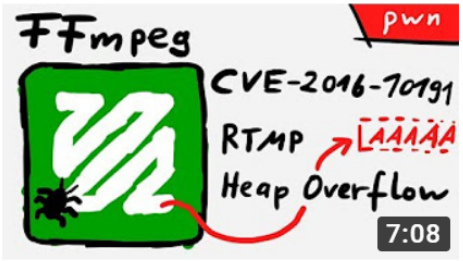
チャンネル登録者数 37,940 人

ホーム　動画　再生リスト　チャンネル　フリートーク　概要



Playing around with a Format String vulnera...

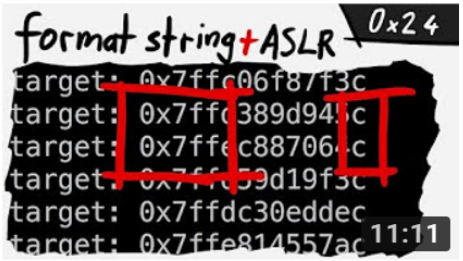
LiveOverflow・視聴回数 2,800 回・6 日前



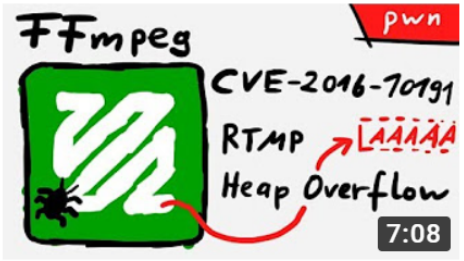
RTMP Heap Overflow CVE-2016-10191 - Exploiting

LiveOverflow
視聴回数 2,189 回・1 週間前

アップロード動画　すべて再生



Playing around with a Format



RTMP Heap Overflow CVE-

Analysis of CVE-2016-10190

First look at a simple PoC

https://www.youtube.com/channel/UClcE-kVhqyiHCcjYwcpfj9w

きっかけの動画

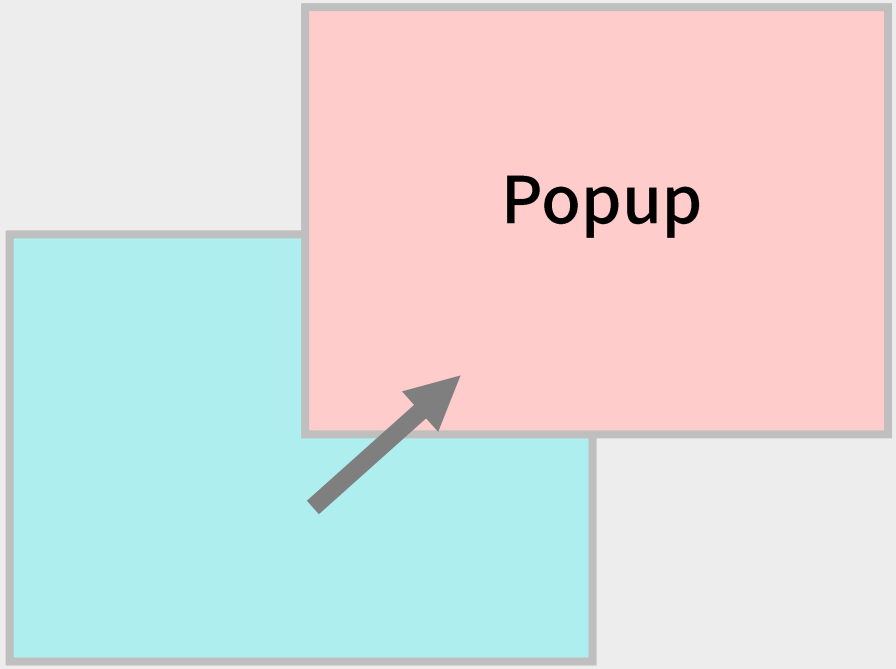
Reverse engineering obfuscated JavaScript - PopUnder Chrome 59
https://youtu.be/8UqHCrGdxOM

Chromeのバグを使ってPreventerをバイパスするライブラリのコード(難読化あり)を読み解き、バグを突き止めている

➡その後、バグ報告&修正！ 👏

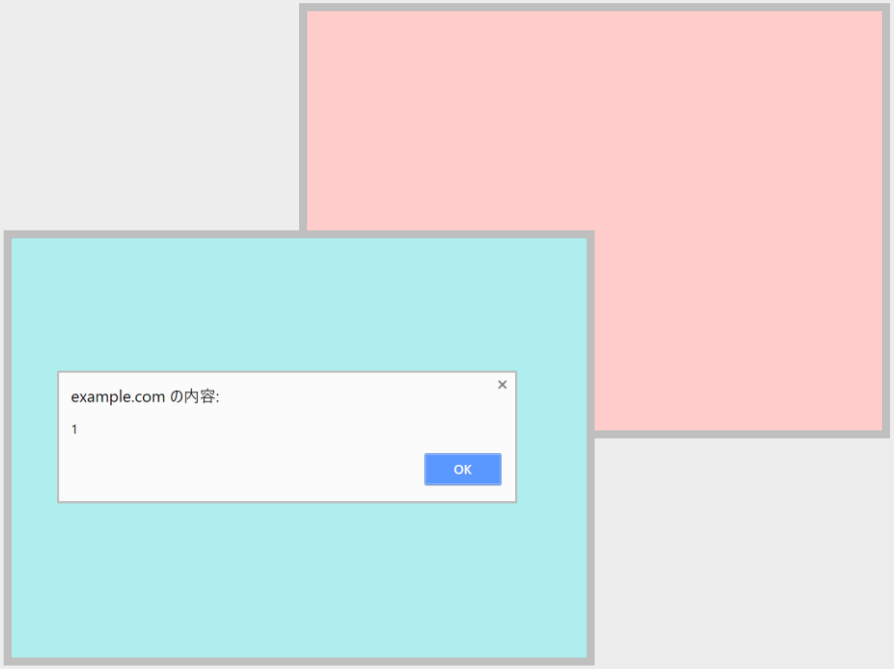
動画中のバイパスのトリック

1. 新しいウインドウ作成



動画中のバイパスのトリック

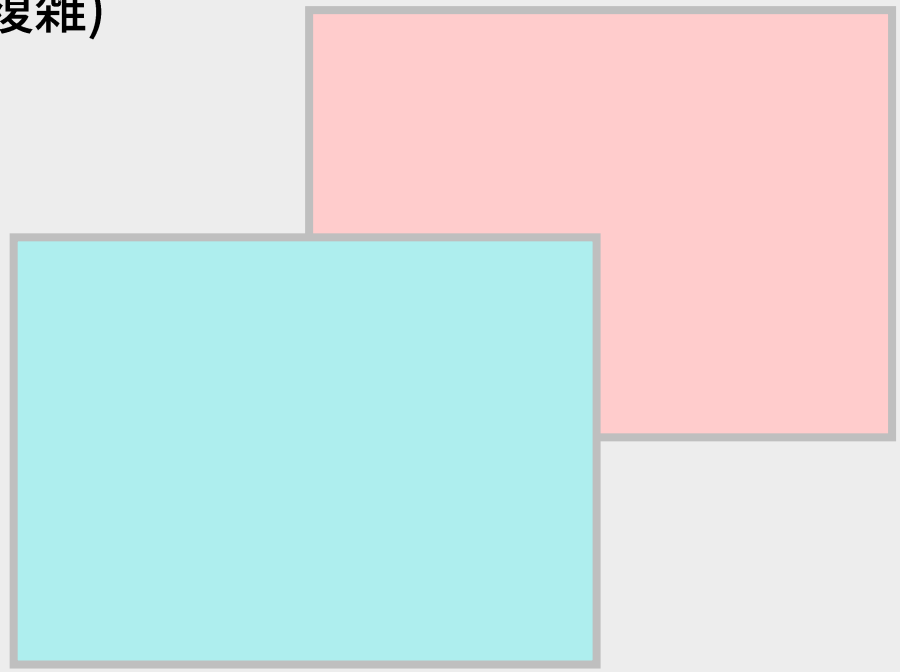
2. 親からalertを出してフォーカスを奪う



動画中のバイパスのトリック

3. alertを消した後もフォーカスは親のまま

(※ 実際はもう少し複雑)



➡ 新しいウインドウは裏へ(Popunderの達成)

自分でも探してみる

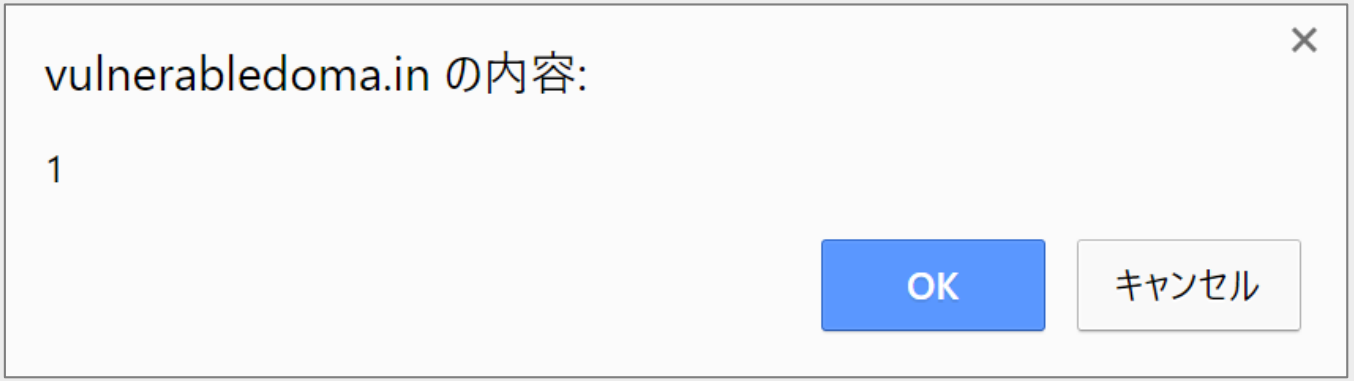
今回はJSで操作するかわりにalertで
フォーカスを奪っていた

alert以外にもフォーカスを奪って
悪用できるダイアログがあるのではないか？

➡調べてみよう

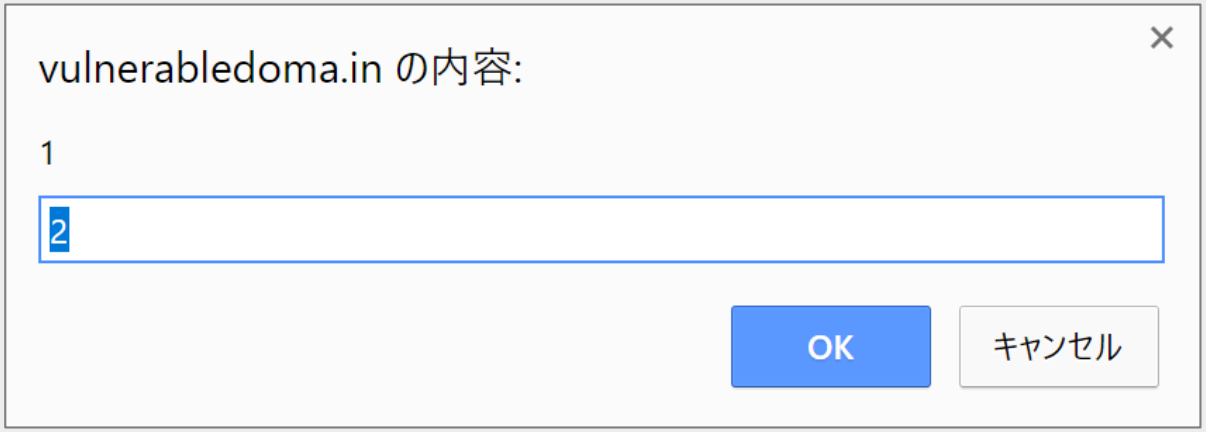
ダイアログいろいろ１

```
confirm(1);
```



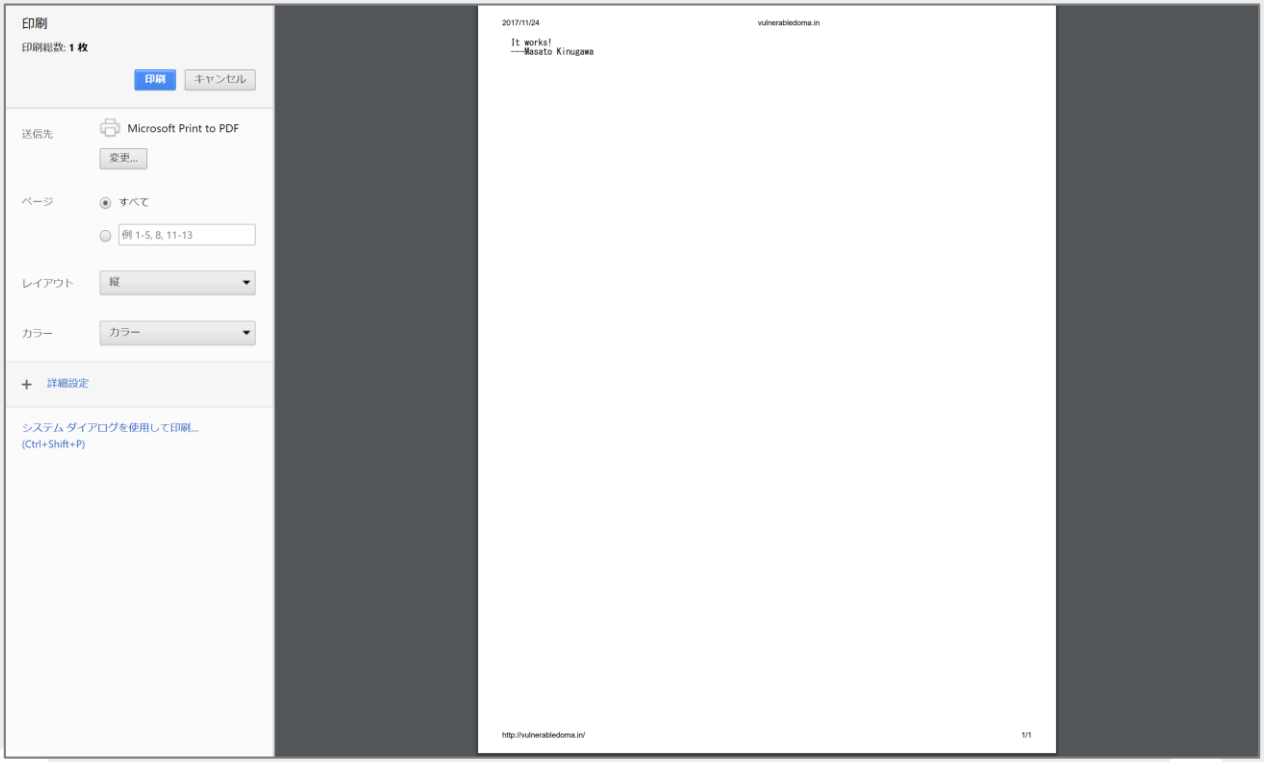
ダイアログいろいろ 2

```
prompt(1);
```



ダイアログいろいろ 3

```
print();
```

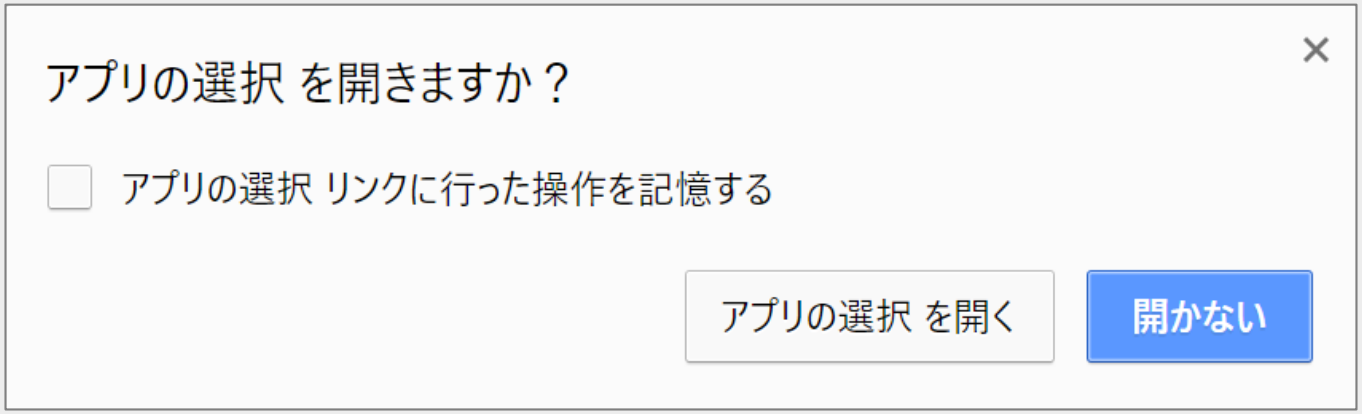


ダイアログいろいろ 4

//Basic認証

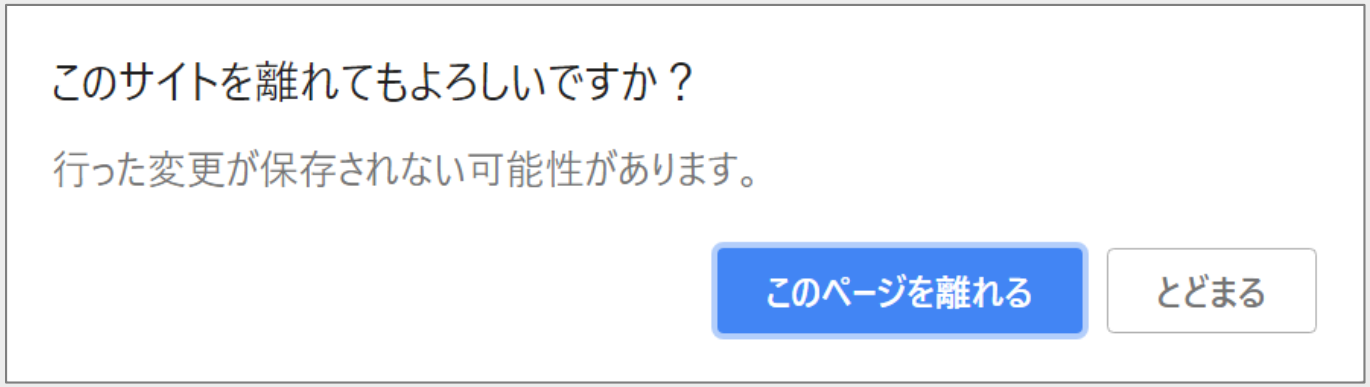
ダイアログいろいろ 5

```
//外部アプリを開くプロトコルへナビゲーション
location = "mms:";
```



ダイアログいろいろ 6

```
onbeforeunload=function(e){return 1;}
//どこかへ移動しようとするとダイアログ出現
```



ダイアログいろいろ 7

```
new PresentationRequest("").start();
```

ダイアログいろいろ 8

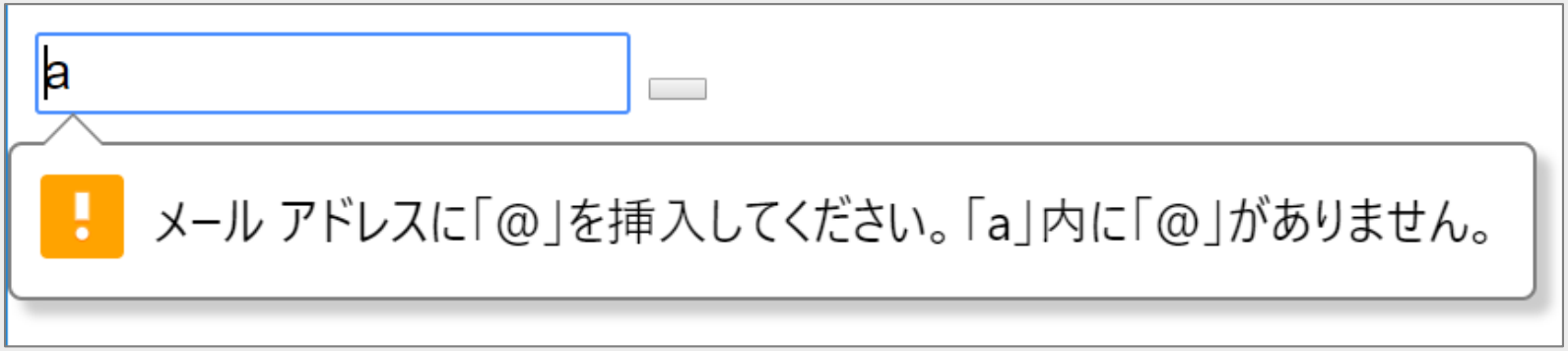
```
new PaymentRequest([{
  supportedMethods: ['basic-card']
}], {
  total: {
    label: 1,
    amount: {
      currency: 'USD',
      value: 0
    }
  }
}).show();
```

ダイアログいろいろ 9

```
navigator.usb.requestDevice({filters:[]});
```

ダイアログいろいろ 10

```
<form>
 <input type="email" value="a">
 <button id="button">
</form>
<script>
  button.click();
</script>
```



テストする

- 手動 & 目視 👀
  - フォーカスを奪うか確認、奪ったらダイアログを自動で消す方法を探る

ダイアログを消す方法の例：

- iframeの中からダイアログを出してiframeごと消す
- ページをリロード
- history.back() するページへ遷移
- CSSでダイアログを出している要素を隠す

うまく動いた例

ダイアログ：Presentation API

消す方法：リロード

```
<script>
function popUnder() {
  new PresentationRequest("").start();
  window.open("https://example.com/", "_blank","a");
  setTimeout(function(){
    location.reload();//リロードでダイアログを消す
  },1000);
}
</script>
<button onclick="popUnder()">Create PopUnder</button>
```

すべての成果 ➡

みつけたもの

Popunder restriction bypass with **payment request API**
https://bugs.chromium.org/p/chromium/issues/detail?id=768230
Popunder restriction bypass with **navigation to external protocol**
https://bugs.chromium.org/p/chromium/issues/detail?id=768475
Popunder restriction bypass with **Presentation API**
https://bugs.chromium.org/p/chromium/issues/detail?id=768900
Popunder restriction bypass with **PDF print() method**
https://bugs.chromium.org/p/chromium/issues/detail?id=769351
Popunder restriction bypass with **form validation error message**(Mac only)
https://bugs.chromium.org/p/chromium/issues/detail?id=769864
Popunder restriction bypass with **PDF confirm dialog**
https://bugs.chromium.org/p/chromium/issues/detail?id=780250

Canary(64)で**すべて修正済み**、 Avi Drissman++!

ダイアログを見て気づいたこと

- それぞれにかなり個性がある
  - フォーカスを強制的に奪うもの
  - ユーザアクション(クリックなど)がないと開けないもの
  - ページのロードを強制的に中断させるもの(print())
  - 手動でしか閉じられないもの
  - ポップアップブロッカーにブロックされるもの(WebUSB)
- 多くがMacとWindowsで動作が違う

➡ UI関連のバグはまだまだ起こりそう！

UIのバグに思うこと

- 比較的ガチャガチャやっていれば発見できる
  - アドレスバー偽装も大体ガチャって発見してきた気がする
  - ブラウザのバグを探したい初心者にもおすすめ！
- ただし、自動化はしにくいと思う
  - 大抵は細かなタイミングや目視の検証がキモになるため

バグを探すときに有効な方法

- 同じ要素を列挙しておくと便利
  - 今回は以前からブラウザのダイアログをみたらメモをとるようにしていたことで効率的に検証することができた

僕のブラウザダイアログコレクションからの1枚：

まとめ

- Popunder Preventerのバイパスをどのように発見したかを紹介しました。
- ダイアログやフォーカス1つも悪用できる！
- UIのバグを探すときはブラウザのダイアログは注目すべき要素かも。

Thanks!